CSCE460301 - Fundamental of Computer Vision (Fall 2024) Instructions

---

- Due date: 11:59pm on Monday, September 30th, 2024
- Submission: through https://canvas.aucegypt.edu
- Starter code and data: Assignment 1 (https://canvas.aucegypt.edu)
- Deliverables:
- Please prepare your answer sheet using the filename of FirstName_LastName_HW1.pdf.
- Include all the code you implement in each problem. Remember to set relative paths so that we can run your code out of the box.
- Compress the code and answer sheet into FirstName_LastName_HW1.zip before uploading it to canvas.

Short programming example (130 points)

Each of the following short programming worth 10 points.

Please use the provided "hokiebird.jpg" as your input.

1) Plot the R, G, B values along the scanline on the 250th row of the image.
   - Save your plot as "01_scanline.png".
2) Stack the R, G, B channels of the *hokiebird* image vertically. This will be an image with width of 600 pixels and height of 1116 pixels.
   - Save the composite image as "02_concat_rgb.png".
3) Load the input color image and swap its red and green color channels.
   - Save the image as "03_swapchannel.png".
4) Convert the input color image to a grayscale image.
   - Save the grayscale image as "04_grayscale.png".
5) Take the R, G, B channels of the image. Compute an average over the three channels. Note that you may need to do the necessary typecasting (uint8 and double) to avoid overflow.
   - Save the averaged image as "05_average.png".
6) Convert the input color image to ycbcr color space.

- Save the *y* component as “06_y_ycbcr.png”.

7) Convert the input color image to a cie_xyz color space.
   - Save the *y* component as “07_y_xyz.png”.
8) Stack the *grayscale*, *y_ycbcr*, *y_xyz* components of the *hokiebird* image horizontally. This will be an image with width of 1800 pixels and height of 372 pixels.
   - Save the composite image as “08_concat_grey.png”.
9) Take the grayscale image in (4), obtain the negative image (i.e., mapping 255 to 0 and 0 to 255).
   - Save the image as “09_negative.png”.
10) First, crop the original *hokiebird* image into a squared image of size 372 x 372. Then, rotate the image by 90, 180, and 270 degrees and stack the four images (0, 90, 180, 270 degrees) horizontally.
   - Save the image as “10_rotation.png”.
11) Create another image with the same size as the *hokiebird* image. First, initialize this image as zero everywhere. Then, for each channel, set the pixel values as 255 when the corresponding pixel values in the *hokiebird* image are greater than 127.
   - Save the image as “11_mask.png”.
12) Report the mean R, G, B values for those pixels marked by the mask in (11).
13) Take the grayscale image in (4). Create and initialize another image as all zeros. For each 5 x 5 window in the grayscale image, find out the maximum value and set the pixels with the maximum value in the 5x5 window as 255 in the new image.
   - Save the result image as “13_nonmax.png”.

You can use either Python or MATLAB for your task.

For MATLAB, some useful functions include:Some useful functions for Matlab: `title, subplot, imshow, mean, imread, imwrite, rgb2gray, find` . Use `help` in MATLAB to learn more about these functions.